H2L How-To Library

Informatica

How to Create Keystore and Truststore Files for Secure Communication in the Informatica Domain

Abstract

When you enable secure communication for the Informatica domain, you secure the connection between services and between the services and the service managers. For better security, you can provide a custom SSL key and certificate. This article describes how to create keystore and truststore files in PEM and JKS formats with a custom SSL key and certificate.

Supported Versions

- Informatica 9.6.x, 10.0, 10.1.x, 10.2.x

Table of Contents

Overview. 2
Creating Keystores and Truststores with a Custom SSL Key and Certificate. 3
Step 1. Create an SSL Key and Certificate. 3
Step 2. Create Keystore Files. 4
Step 3. Create Truststore Files. 5
Enabling Secure Communication with Keystore and Truststore Files. 5
Before Enabling Secure Communication. 5
Enabling Secure Communication for the Domain. 6

Overview

To secure the connection between services and between the services and the service managers in the Informatica domain, use the secure communication option.

When you enable secure communication, you secure the connections between the following components:

- The Service Manager and all services running in the domain
- The Data Integration Service and the Model Repository Service
- The Data Integration Service and the workflow processes
- The PowerCenter Integration Service and the PowerCenter Repository Service
- The domain services and the Informatica client tools and command line programs

To enable secure communication in the domain, you need an SSL key and certificate. SSL keys and certificates are stored in keystore and truststore files.

Informatica provides keystores and truststores based on an SSL key and certificate that is common to all Informatica installations. For better security, you can provide a custom SSL key and certificate. The SSL certificate can be self-signed or issued by a certificate authority. Use a certificate from a certificate authority for a more secure domain.

Informatica requires keystore and truststore files in PEM and JKS formats with the following names:

- infa_keystore.pem
- infa_truststore.pem
- infa_keystore.jks
- infa_truststore.jks

If you have a keystore and truststore in JKS format, export the certificate and key to create the keystore and truststore in PEM format.

If you have a keystore and truststore in PEM format, convert the PEM keystore file to PKCS12. Then, export the certificate and key to JKS files.

If you do not have keystore and truststore files, you can create them with OpenSSL and Java keytool.

Download OpenSSL at the following link: http://www.openssl.org/source/.

Java keytool is part of the Java Development Kit (JDK). Download the JDK at the following link: http://www.oracle.com/technetwork/java/javase/downloads/index.html

The software available for download at the referenced links belongs to a third party or third parties, not Informatica Corporation. The download links are subject to the possibility of errors, omissions or change. Informatica assumes no responsibility for such links and/or such software, disclaims all warranties, either express or implied, including but not limited to, implied warranties of merchantability, fitness for a particular purpose, title and non-infringement, and disclaims all liability relating thereto.

Creating Keystores and Truststores with a Custom SSL Key and Certificate

When you enable secure communication for the domain, Informatica requires keystores and truststores. Create the keystores and truststores with a custom SSL key and certificate for a higher level of security.

To create keystore and truststore files in PEM and JKS formats on a UNIX or Linux operating system, perform the following steps:

1. Create an SSL key and certificate.
2. Create keystore files.
3. Create truststore files.

You need OpenSSL and keytool to complete the steps.

Step 1. Create an SSL Key and Certificate

Create an SSL key and certificate with OpenSSL.

1. Create a SSL key and certificate signing request (CSR).

   Run the following command:

   ```
   $ openssl req -new -newkey <encryption algorithm>:<number of bits> -<digest> -keyout
   <key file> -out <CSR file>
   ```

   For example, the following command uses 2048 bit RSA encryption and SHA1 digest to create a key file named keystore.key and a CSR file named keystore.csr:

   ```
   $ openssl req -new -newkey rsa:2048 -sha1 -keyout keystore.key -out keystore.csr
   ```

2. Provide the requested information to create the key.

The following image shows the requested information:

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Your State
Locality Name (eg, city) []:Your City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your Organization
Organizational Unit Name (eg, section) []:Your Section
Common Name (e.g. server FQDN or YOUR name) []:Your Common Name
Email Address []:Email@EmailAddress.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Challenge Password
An optional company name []:
```

3. Create a certificate.

   Run the following command:

   ```
   $ openssl x509 -req -days <number of days> -in <CSR file> -signkey <key file> -out <CRT file>
   ```

   The command creates a self-signed certificate. Use a certificate from a certificate authority for a higher level of security.

   For example, the following command uses a CSR file named keystore.csr and a key file named keystore.key to create a self-signed certificate named keystore.crt that is valid for 11499 days:

   ```
   $ openssl x509 -req -days 11499 -in keystore.csr -signkey keystore.key -out keystore.crt
   ```

Step 2. Create Keystore Files

Create keystores in PEM and JKS formats with a CRT file and a key file. The keystore files must contain the root and intermediate SSL certificates.

1. Create the keystore file in PEM format.

   Run the following command:

   ```
   $ cat <CRT file> <key file> >> <PEM keystore>
   ```

   For example, the following command uses a CRT file named keystore.crt and a key file named keystore.key to create a PEM keystore named infa_keystore.pem:

   ```
   $ cat keystore.crt keystore.key >> infa_keystore.pem
   ```

   Note: The keystore in PEM format must be named "infa_keystore.pem" and is case sensitive.

2. Convert the keystore in PEM format to PKCS12 format.

   Run the following command:

   ```
   $ openssl pkcs12 -export -in <PEM keystore> -out <PKCS12 keystore> -name <name>
   ```

   For example, the following command uses a PEM keystore named infa_keystore.pem to create a PKCS12 keystore named keystore.p12 with the name informatica for the certificate and private key:

   ```
   $ openssl pkcs12 -export -in infa_keystore.pem -out keystore.p12 -name "informatica"
   ```

3. Convert the keystore in PKCS12 format to JKS format.

Run the following command:

```
$ keytool -v -importkeystore -srckeystore <PKCS12 keystore> -srcstoretype PKCS12 -
destkeystore <JKS keystore> -deststoretype JKS -srcalias <alias> -destalias <alias>
```

For example, the following command converts a keystore in PKCS12 format named keystore.p12 to a keystore in JKS format named infa_keystore.jks from a source named informatica to a destination named informatica:

```
$ keytool -v -importkeystore -srckeystore keystore.p12 -srcstoretype PKCS12 -
destkeystore infa_keystore.jks -deststoretype JKS -srcalias "informatica" -destalias
"informatica"
```

Note: The keystore in JKS format must be named "infa_keystore.jks" and is case sensitive.

The password for the keystore in JKS format must be the same as the private key pass phrase used to generate the SSL key.

Step 3. Create Truststore Files

Create truststores in PEM and JKS formats with a CRT file and keytool. The truststore files must contain the root, intermediate, and end user SSL certificates.

1. Create a truststore in PEM format.

   Run the following command:

   ```
   $ cat <CRT file> >> <PEM truststore>
   ```

   For example, the following command uses a CRT file named keystore.crt to create a truststore in PEM format named infa_truststore.pem:

   ```
   $ cat keystore.crt >> infa_truststore.pem
   ```

   Note: The truststore in PEM format must be named "infa_truststore.pem" and is case sensitive.

2. Create a truststore in JKS format.

   Run the following command:

   ```
   $ keytool -importcert -file <PEM truststore> -keystore <JKS truststore> -alias "<alias>"
   deststoretype JKS -v -trustcacerts
   ```

   For example, the following command uses a PEM truststore named infa_truststore.pem to create a truststore in JKS format named infa_truststore.jks with the alias informatica:

   ```
   $ keytool -importcert -file infa_truststore.pem -keystore infa_truststore.jks -alias
   "informatica" deststoretype JKS -v -trustcacerts
   ```

   Note: The truststore in JKS format must be named "infa_truststore.jks" and is case sensitive.

Enabling Secure Communication with Keystore and Truststore Files

After you create the keystore and truststore files, use the files to enable secure communication for the domain.

Before Enabling Secure Communication

Before you enable secure communication for the domain, the Informatica installation needs access to the keystore and truststore files.

Move the following files to a directory accessible to the installer, the Administrator tool, or the command line programs:

- infa_keystore.jks
- infa_keystore.pem

- infa_truststore.jks
- infa_truststore.pem

You need the following information:

- The directory location of the keystore and truststore files
- The password for infa_keystore.jks
- The password for infa_truststore.jks

Enabling Secure Communication for the Domain

You can enable secure communication when you install Informatica or after you install Informatica.

To enable secure communication for the domain, use one of the following methods:

- During installation, select the **Enable secure communication for the domain** option.
- After installation, edit the **General Properties** for the domain in the Administrator tool and run the infasetup UpdateGatewayNode or UpdateWorkerNode command.
- After installation, run the infacmd isp UpdateDomainOptions command and the infasetup UpdateGatewayNode or UpdateWorkerNode command.

For more information about enabling secure communication, see the *Informatica Security Guide*.

Author

Brian Le
Technical Writer